

04860.P2207

UNITED STATES PATENT APPLICATION

FOR

METHOD AND APPARATUS FOR MEDIA DATA TRANSMISSION

INVENTORS:

ANNE JONES
JAY GEAGAN
KEVIN L. GONG
ALAGU PERIYANNAN
DAVID W. SINGER

PREPARED BY:

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN
12400 WILSHIRE BOULEVARD
SEVENTH FLOOR
LOS ANGELES, CA 90025-1026

(408) 720-8598

EXPRESS MAIL CERTIFICATE OF MAILING

"Express Mail" mailing label number: EM492088673US

Date of Deposit: August 25, 1998

I hereby certify that I am causing this paper or fee to be deposited with the United States Postal Service "Express Mail Post Office to Addressee" service on the date indicated above and that this paper or fee has been addressed to the Assistant Commissioner for Patents, Washington, D. C. 20231

Connie Thayer
(Typed or printed name of person mailing paper or fee)

Connie Thayer
(Signature of person mailing paper or fee)

8-25-98
(Date signed)

INS. 01

METHOD AND APPARATUS FOR MEDIA DATA TRANSMISSION

FIELD OF THE INVENTION

5 The present invention relates to methods and apparatuses for preparing time related sequences of media data for transmission, and more particularly to packetized transmission of such media data.

INTRODUCTION AND BACKGROUND

10 There are various different file structures used today to store time-based media: audio formats such as AIFF, video formats such as AVI, and streaming formats such as RealMedia. One reason that such file structures are different is their different focus and applicability. Some of these formats are sufficiently relatively widely accepted, broad in their application, and somewhat simple to implement, and thus, may be used
15 not only for content delivery but also as interchange formats. Foremost among these general formats is the QuickTime file format. It is used today in the majority of web sites serving time-based data; in the majority of authoring environments, including professional ones; and on the majority of multimedia CDROM titles.

 The QuickTime media layer supports the efficient display and management of
20 general multimedia data, with an emphasis on time-based material (video, audio, etc.). The media layer uses the QuickTime file format as the storage and interchange format for media information. The architectural capabilities of the layer are generally broader than the existing implementations, and the file format is capable of representing more information than is currently demanded by the existing QuickTime implementations.

In contrast to formats such as AVI, which were generally designed to support *local* random access of synchronized media, QuickTime allows systems to manage the data, relationships and timing of a general multimedia presentation. In particular, the QuickTime file format has structures to represent the temporal behavior of general
5 time-based streams, a concept which covers the time-based emission of network packets, as well as the time-based local presentation of multimedia data.

The existing QuickTime file format is publicly described by Apple Computer in the May 1996 File format specification, which may be found at the QuickTime site, <http://www.apple.com/quicktime>.

10 One aspect of the QuickTime file format is the concept that the physical structure of media data (the layout in disk records) is independent of, and described by, a logical structure for the file. The file is fully described by a set of "movie" meta-data. This meta-data provides declarative, structural and temporal information about the actual media data.

15 The media data may be in the same file as the description data, (the "movie" meta-data), or in other file(s). A movie structured into one file is commonly called "flat", and is self-contained. Non-flat movies can be structured to reference some, or all, of the media data in other files.

As such, the format is generally suited for optimization in different
20 applications. For example, when editing (compositing), data need not be rewritten as edits are applied and media is re-ordered; the meta-data file may be extended and temporal mapping information adjusted. When edits are complete, the relevant media data and meta-data may be rewritten into a single, interleaved, and optimized file for

local or network access. Both the structured and the optimized files are valid QuickTime files, and both may be inspected, played, and reworked.

The use of structured ("non-flat") files enables the same basic media data to be used and re-used in any number of presentations. This same advantage applies when
5 serving, as will be seen below.

In both editing and serving, this also permits a number of other files to be treated as part of a movie without copying the media data. Thus editing and serving may be done directly from files such as Sun Microsystem's "au" audio format or the AVI video format, greatly extending the utility of these formats.

10 The QuickTime file is divided into a set of objects, called atoms. Each object starts with an atom header, which declares its size and type:

```
class Atom {  
    int(32)      size;  
    char         type[4];  
15    byte       contents[];  
}
```

The size is in bytes, including the size and type header fields. The type field is four characters (usually printable), to permit easy documentation and identification. The data in an object after the type field may be fields, a sequence of contained
20 objects, or both.

A file therefore is simply a sequence of objects:

```
class File {  
    Atom[];  
}
```

- 5 The two important top-level objects are the media-data (mdat) and the meta-data (moov).

10 The media-data object(s) contain the actual media (for example, sequences of sound samples). Their format is not constrained by the file format; they are not usually objects. Their format is described in the meta-data, not by any declarations physically contiguous with them. So, for example, in a movie consisting solely of motion-JPEG, JPEG frames are stored contiguously in the media data with no intervening extra headers. The media data within the media data objects is logically divided into chunks; however, there are no explicit chunk markers within the media data.

- 15 When the QuickTime file references media data in other files, it is not required that these 'secondary' files be formatted according to the QuickTime specification, since such media data files may be formatted as if they were the contents of a media object. Since the QuickTime format does not necessarily require any headers or other information physically contiguous with the media data, it is possible for the media data
- 20 to be files which contain 'foreign' headers (e.g. UNIX ".au" files, or AVI files) and for the QuickTime meta-data to contain the appropriate declarative information and reference the media data in the 'foreign' file. In this way the QuickTime file format can be used to update, without copying, existing bodies of material in disparate
- 25 formats. The QuickTime file format is both an established format and is able to work with, include, and thereby bring forward, other established formats.

Free space (e.g. deleted by an editing operation) can also be described by an object. Software reading a file that includes free space objects should ignore such free space objects, as well as objects at any level which it does not understand. This permits extension of the file at virtually any level by introducing new objects.

- 5 The primary meta-data is the movie object. A QuickTime file has exactly one movie object which is typically at the beginning or end of the file, to permit its easy location:

```
10       class Movie {  
          int(32)       size;  
          char         type[4] = 'moov';  
          MovieHeader mh;  
          contents     Atom[];  
      }
```

- 15 The movie header provides basic information about the overall presentation (its creation date, overall timescale, and so on). In the sequence of contained objects there is typically at least one track, which describes temporally presented data.

```
20       class Track {  
          int(32)       size;  
          char         type[4] = 'trak';  
          TrackHeader th;  
          contents     Atom[];  
      }
```

- 25 The track header provides relatively basic information about the track (its ID, timescale, and so on). Objects contained in the track might be references to other tracks (e.g. for complex compositing), or edit lists. In this sequence of contained objects there may be a media object, which describes the media which is presented when the track is played.

The media object contains declarations relating to the presentation required by the track (e.g. that it is sampled audio, or MIDI, or orientation information for a 3Dscene). The type of track is declared by its handler:

```
5  class handler {
    int(32)      size;
    char         type[4] = 'hdlr';
    int(8)       version;
    bit(24)      flags;
10  char         handlertype[4];  -- mhlr for media handlers
    char         handlersubtype[4] -- vide for video, soun for
audio
    char         manufacturer[4];
    bit(32)      handlerflags;
    bit(32)      handlerflagsmask;
15  string       componentname;
    }
```

Within the media information there is likewise a handler declaration for the data handler (which fetches media data), and a data information declaration, which defines which files contain the media data for the associated track. By using this
20 declaration, movies may be built which span several files.

At the lowest level, a sample table is used which relates the temporal aspect of the track to the data stored in the file:

```
class sampletable {
25  int(32)      size;
    char         type[4] = 'stbl';
    sampleddescription sd;
    timetosample  tts;
    syncsampletable syncs;
    sampletochunk stoc;
30  samplesize   ssize;
    chunkoffset  coffset;
    shadowsync   ssync;
    }
```

The sample description contains information about the media (e.g. the
35 compression formats used in video). The time-to-sample table relates time in the track, to the sample (by index) which should be displayed at that time. The sync

sample table declares which of these are sync (key) samples, not dependent on other samples.

The sample-to-chunk object declares how to find the media data for a given sample, and its description given its index:

```
5  class sampletochunk {  
    int(32)      size;  
    char         type[4] = 'stsc';  
    int(8)       version;  
    bits(24)     flags;  
10  int(32)      entrycount;  
    for (int i=0; i<entrycount; i++) {  
        int(32)   firstchunk;  
        int(32)   samplesperchunk;  
        int(32)   sampledescriptionindex;  
15  }  
}
```

The sample size table indicates the size of each sample. The chunkoffset table indicates the offset into the containing file of the start of each chunk.

Walking the above-described structure to find the appropriate data to display for a given time is fairly straightforward, generally involving indexing and adding. Using the sync table, it is also possible to back-up to the preceding sync sample, and roll forward 'silently' accumulating deltas to a desired starting point.

Figure 1 shows the structure of a simple movie with one track. A similar diagram may be found in the QuickTime file format documentation, along with a detailed description of the fields of the various objects. QuickTime atoms (objects) are shown here with their type in a grey box, and a descriptive name above. This movie contains a single video track. The frames of video are in the same file, in a single chunk of data. It should be noted that the 'chunk' is a logical construct only; it is not an object. Inside the chunk are frames of video, typically stored in their native form. There are no required headers or fields in the video frames themselves.

Figure 2 is a diagram of a self-contained file with both an audio and a video track. Fewer of the atoms are shown here, for brevity; the pointers from the tracks into the media data are, of course, the usual sample table declarations, which include timing information.

5 The QuickTime file format has a number of advantages, including:

- 10 1) Scalability for size and bit-rates. The meta data is flexible, yet compact. This makes it suitable for small downloaded movies (e.g. on the Internet) as well as providing the basis for a number of high-end editing systems.
- 15 2) Physical structure is independent of the logical and temporal structure. This makes it possible to optimize the physical structure differently depending on the use the file will have. In particular, it means that a single file format is suitable for authoring and editing; downloading or placing on CDROMs; and for streaming.
- 20 3) The file format has proven capable of handling a very broad variety of codec types and track types, including many not known at the time the format was designed. This proven ability to evolve in an upwards-compatible fashion is fundamental to the success of a storage format.

Scalable, or layered, codecs can be handled in a number of ways in the QuickTime file format. For a streaming protocol which supports scalability, the samples may be tagged with the layer or bandwidth threshold to be met for
25 transmitting the samples.

Tracks which form a set of alternatives (e.g. different natural language sound tracks) can be tagged so that only one is selected for playback. The same structure can be used to select alternatives for streaming (e.g. for language selection). This capability is described in further detail in the QuickTime file format.

30 When QuickTime displays a movie or track, the appropriate media handler accesses the media data for a particular time. The media handler must correctly

interpret the data stream to retrieve the requested data. For example, with respect to video media, the media handler typically traverses several atoms to find the location and size of a sample for a given media time. The media handler may perform the following:

5

1. Determine the time in the media time coordinate system.
2. Examine the time-to-sample atom to determine the sample number that contains the data for the specified time.
- 10 3. Scan the sample-to-chunk atom to discover which chunk contains the sample in question.
4. Extract the offset to the chunk from the chunk offset atom.
- 15 5. Find the offset within the chunk and the sample's size by using the sample size atom.

It is often desirable to transmit a QuickTime file or other types of time related sequences of media data over a data communication medium, which may be associated with a computer network (e.g. the Internet). In many computer networks, the data which is transmitted into the network should generally be in a packet form. Normally, time related sequences of media data are not in the proper packetized format for transmission over a network. For example, media data files in the QuickTime format are not in a packetized format. Thus, there exists a need to collect the data, sometimes referred to as streaming data, into packets for transmission over a network.

One prior approach to address the problem of transmitting time related sequences of media data over a network is to send the media file over the network using a network or transmission protocol, such as the Hypertext Transfer Protocol (HTTP). Thus, the media file itself is sent from one computer system over the network to another computer system. However, there may be no desire to retain the

30

media file at the receiving computing system. That is, when the media file is received and viewed or listened to at the receiving computer system, there may be no desire by the user of that receiving computer system to store a copy of the file, for example, if the receiving computing system is a network computer or a computer with low storage capacity.

Another alternative approach to solving the problem of how to collect data for transmission by packets over a network is to prepare a file which contains the network protocol data units in the file for a particular transmission protocol. In a sense, such a file may be considered a packetized file which is stored in essentially the same format as it will be transmitted according to the particular transmission protocol. Performing this operation generally involves storing the file in a packetized form for a particular network protocol at a particular data transmission rate and a particular media file format. Thus, for each different transmission protocol at a particular data transmission rate, the file will essentially be replicated in its packetized form. The fixed form of such files may restrict their applicability/compatibility and make it difficult to view such files locally. Thus, such an approach may greatly increase storage requirements in attempting to provide the file in various transmission protocols at various different data transmission rates. Moreover, each packetized file generated according to this alternative prior approach is generally limited to a particular media file format, and thus, other media file formats for the same media object (e.g. a digital movie) are typically packetized and stored on the sending computer system.

Yet another approach to solving the problem of how to stream time related sequences of media data is to perform the packetization of the media data when

required on the transmitting system according to the particular transmission protocol which is desired. This processing requires, in many cases, a relatively considerable amount of time, and thus, may slow the performance of the transmitting system.

Thus, it is desirable to provide an improved method and apparatus for
5 transmitting time related sequences of media data.

10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
222

SUMMARY OF THE INVENTION

The present invention provides methods and apparatuses for processing media data for transmission in a data communication medium. In one embodiment, a set of data indicates how to transmit a time related sequence of media data according to a transmission protocol. The set of data, according to one embodiment, includes a time related sequence of data which is associated with the time related sequence of media data. According to one aspect of the invention, the set of data may be utilized by a digital processing system to transmit the time related sequence of media data (e.g., by packets generated according to the transmission protocol and the set of data).

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 shows an example of the structure of a simple movie with one track in the prior art.

Figure 2 is an example of a self-contained movie file of the prior art.

5 Figure 3 is a flowchart showing one example of a method according to the present invention.

Figure 4 shows an example of a hint track of the present invention.

Figure 5 shows another example of a hint track of the present invention.

10 Figure 6 is a diagram of a network of computer systems in which media data may be exchanged and/or processed, according to one embodiment of the present invention.

Figure 7 is a block diagram of a digital processing system which may be used in accordance with one embodiment of the present invention.

15 Figure 8 is a block diagram of a system that utilizes hints to transfer media data, according to one embodiment of the invention.

Figure 9 is a block diagram of a system that utilizes hints to transfer media data, according to one embodiment of the invention.

Figure 10 is a flow diagram illustrating a method for generating hints for providing media data transmission, according to one embodiment of the invention.

20 Figure 11 is a flow diagram illustrating a method of processing media data received by a receiving system in accordance with hints, according to one embodiment of the invention.

Figure 12 is an example of a machine readable storage medium that may be accessed by a digital processing system, such as a generator, according to one embodiment of the invention.

Figure 13 is an example of a machine readable storage medium that may be
5 accessed by a digital processing system, such as a server, according to one embodiment of the invention.

Figure 14 is an example of a machine readable storage medium that may be accessed by a digital processing system, such as a receiving system or other digital processing system, according to one embodiment of the invention.

10 Figure 15 is a diagram of a data storage and/or communication medium having stored/transported thereon media and hint information, according to one embodiment of the invention.

DETAILED DESCRIPTION

The present invention provides methods and apparatuses for allowing the transmission, and particularly the packetized transmission of time related sequences of media data, which may include, for example, video, audio, video and audio, etc.,
5 over a communication media, such as in a computer network.

In one embodiment of the present invention, a digital processing system creates a set of data for indicating how to transmit a time related sequence of media data according to a transmission protocol. Typically, this set of data is stored on a storage device coupled to the digital processing system. Further, this set of data is a
10 time related sequence of data associated with the time related sequence of media data.

The present invention may be implemented entirely in executable computer program instructions which are stored on a computer readable media or may be implemented in a combination of software and hardware, or in certain embodiments, entirely in hardware. Typically, a server computer system coupled to a network will create the set of data, which may be referred to as a hint track and will store this hint
15 track in a storage device which is coupled to the server computer system. When a client computer system requests a presentation (e.g. a viewing or listening or viewing and listening) of a media data file, the server system uses the hint track to determine how to packetize the media data for transmission to the client computer system. It will
20 be appreciated that the present invention is generally applicable to time related sequences of media data, and that QuickTime is represented herein as one example of this general applicability. Thus, the invention should not necessarily be limited to QuickTime.

Figure 3 shows one example of a method according to the present invention. The method 300 shown in Figure 3 begins in step 301, in which the media file format for the particular media data which is desired to be transmitted is determined. In step 303, the particular transmission protocol or protocols which are desired to be used is also determined. However, steps 301 and 303 are optional, for example, in the case where the same media file format is always transmitted using the same transmission protocol.

In step 305, a digital processing system, such as a server computer system, creates and stores the hints for packetizing a time related sequence of media data in a media file. Alternatively, one computer system may create the hints and provide them to another system, such as a server computer system, which stores them for later use in a transmission process. The packetization allows the transmission over a network or communication media according to the desired transmission protocol which was determined in step 303. In one embodiment of the present invention, the hints are stored as a track of time related sequence of hints which refers to, but which in one embodiment, is separate from other tracks of media data. The track of hints, in one embodiment of the present invention, may be stored separately from the media data to which it refers. As such, the track of hints may be stored in a file which is distinct from another file containing the media data which is referred to by the track of hints, or the track of hints may be stored in a hint area in the file containing the media data which is separate and distinct from the data area containing the actual media data. In one embodiment of the invention, a hint track, or portion thereof, may be interpreted as executable instructions by the server, which executable instructions cause the server

to packetize a time related sequence of data, which is typically, but not necessarily, time-based media data. In one embodiment of the present invention, the hints are stored on the storage device which is coupled to the transmitting digital processing system.

5 In step 307, the data which is packetized according to the hints, is transmitted from a transmitting system, such as a server computer system, to a receiving system. This media data is transmitted by packetizing the media data according to the hints. In one alternative embodiment of the invention, the server computer system may decide not to use the hints and to send the media data by an alternative packetization process.

10 In step 309, the receiving system presents the media object which is represented by the media data. Typically, this presentation (which may be a viewing and listening of a media object or merely a viewing or merely a listening of the media object) is performed as the packetized data is received at the receiving system. The packetized data may, in one embodiment of the present invention, but need not be,
15 stored on the receiving system. Thus the presentation of the data is ephemeral in the sense that once the presentation is over, there is no local copy at the receiving system. In another embodiment, presentation of the media object may take place on the server system subsequent to creating hints for the media data representing the media object. In one embodiment of the invention, the media data is not necessarily (re)formatted,
20 copied, etc., for packetization according to hints.

 In step 311, the receiving system may optionally reassemble the media file if the media file as received has been stored on the receiving system. It will be appreciated that the various steps of the method shown in Figure 3 may be performed

in a different order than the one shown and described above and/or some of the steps may be performed simultaneously. For example, in one embodiment, steps 309 and 311 are performed in parallel.

A particular implementation with QuickTime according to one embodiment of
5 the present invention will now be described. In one embodiment of the present invention, a presentation which can be both viewed locally to the file (e.g., at a server, generator, etc.), and streamed over a network within a QuickTime movie is provided. In general, the streaming server (or another system) should have information about the data units to stream, their composition and timing. Since such information is typically
10 temporal it may be described in tracks. A server may perform packetization and determine protocol information, for example, by using the same indexing operations as would be used to view a presentation.

The tracks which contain instructions for the servers are sometimes referred to as 'hint' tracks, since such tracks represent a set of data to direct the server in the
15 process of forming and transmitting packets. The QuickTime file format supports streaming of media data over a network as well as local playback. The process of sending protocol data units is time-based, just like the display of time-based data, and is therefore suitably described by a time-based format. A QuickTime file or 'movie' which supports streaming includes information about the data units to stream. This
20 information is included in additional tracks of the file called "hint" tracks.

Hint tracks contain instructions for a streaming server (or other digital processing system) which assist in the formation of packets. These instructions may contain immediate data for the server to send (e.g. header information) or reference

segments of the media data. In one embodiment of the present invention, instructions are encoded in the QuickTime file in the same way that editing or presentation information is encoded in a QuickTime file for local playback. Instead of editing or presentation information, information may be provided which may allow a server to
5 packetize the media data in a manner suitable for streaming using a specific network transport.

In one embodiment of the present invention, the same media data is used in a QuickTime file which contains hints, whether it is for local playback, or streaming over a number of different transport types. Separate 'hint' tracks for different
10 transport types may be included within the same file and the media may play over all such transport types without making any additional copies of the media itself. In addition, existing media may be made streamable by the addition of appropriate hint tracks for specific transports. According to one aspect of the invention, media data itself need not be recast or reformatted.

15 Therefore the samples in a hint track generally contain instructions to form packets. These instructions may contain immediate data for the server to send (e.g. header information) or reference segments of the media data in another track.

In one embodiment of the present invention, a three-level design is utilized such that:
20

- 21 1) The media data is represented as a set of network-independent tracks, which may be played, edited, and so on, as normal;
- 25 2) There is a common declaration and base structure for server hint tracks; this common format is protocol independent, but contains the declarations of which protocol(s) are described in the server track(s);

- 3) There is a specific design of the server hint tracks for each protocol which may be transmitted; all these designs use the same basic structure. For example, there may be designs for RTP (for the Internet) and MPEG-2 transport (for broadcast), or for new standard or vendor-specific protocols.

5

In one embodiment of the present invention, the resulting streams, sent by the servers under the direction of the hint tracks, are normal streams, and do not necessarily include a trace of QuickTime information. This embodiment of the invention does not require that QuickTime, or its structures or declaration style, necessarily be either in the data on the transmission medium (e.g. network cable) or in the decoding station. For example, a file using H.261 video and DVI audio, streamed under RTP, may result, in one embodiment of the present invention, in a packet stream which is fully compliant with the IETF specifications for packing those codings into RTP.

15

In one embodiment of the invention, hint tracks are built and flagged so that when the presentation is viewed locally, the hint tracks are essentially ignored by a receiving system.

In one embodiment, a time related sequence of media data, which may, for example, include video, audio, etc., may be packetized by a digital processing system, and then presented on the same digital processing system. Furthermore, packetization may be ephemeral, such that the time related sequence being presented, stored, read, etc., is also packetized "on the fly." In one embodiment, hints may refer to media data that has not been copied, formatted, etc.; for example, the media data to which hints refer may be stored in original format on a read-only memory, etc.

25

In one embodiment, the same hinting routine that provides packetization also presents the media as packetization is performed. In alternative embodiments of the invention, a packetized file of time related media data may be generated according to hint tracks and stored, for example, for later transmission.

- 5 Figure 4 illustrates utilization of hint tracks for transporting media data, according to one embodiment of the invention. In Figure 4, a hint track 401 is shown for the media track 403. Each hint track sample, such as hint track sample 405— which describes how to form an RTP packet—may contain a header, and may reference some data from an associated media track—in this case, a video track 403.
- 10 In the embodiment shown in Figure 4, the media data (the video frames) and the RTP hints have been interleaved so that the associated media file may be read relatively easily. In this example, each frame is shown as fitting into a single RTP packet. Of course, it is possible to split frames into several packets when needed. Conversely, multiple frames can, if desired, be placed in a single packet, which is commonly
- 15 performed with audio data.

As discussed above, the logical structure described above need not imply physical structure. The meta data may be cached in memory, and the hint track samples physically interleaved with the media samples to which they refer (as is shown in Figure 4).

- 20 Alternatively, it is possible to write a new set of meta data and media data, containing the hint tracks, which references and augments the meta data and media data in an existing presentation. Figure 5 illustrates utilization of hint tracks to reference media data in a separate file, according to one embodiment of the invention.

In Figure 5, two movie files 502 and 504 are shown, each with their own meta-data. The first, the movie file 502, includes a video track. The second, the movie file 504, contains both a video track and a hint track, but the meta-data declares that the media data for the video track is in the first movie 502. Thus the hints associated with the
5 movie file 504 also point to the media data in the first movie 502.

In one embodiment of the present invention, a media file may contain packetization hint tracks for multiple protocols. As such, each track may contain declarations of the protocol (and protocol parameters, if appropriate) for which the hint track is appropriate. These tracks may all, of course, reference media data from
10 the basic media tracks in the file. The desire for protocol independence and extensibility may be met in the described manner.

In one embodiment of the present invention, hint tracks need not use all the data in the media tracks. The hint tracks may use a subset of the data (e.g. by omitting some video frames) to reach a bandwidth threshold, or for other reasons. Since
15 multiple hint tracks may be provided for the same protocol, differing subsets of the same basic media information at different rates may be provided. As such, the present invention may provide improved scalability over prior methods and apparatuses.

It should be emphasized that though the hint tracks themselves, and the QuickTime meta-data, should, in one embodiment, be in QuickTime files, the base
20 media can be left in any file type which QuickTime can import and reference in place. In one embodiment of the present invention, the meta-data in the movie file may include a data reference which declares that the media data is in another file. The sample table offsets and pointers may thus refer to data in this 'foreign' file. Thus,

according to one embodiment of the present invention, existing legacy formats such as “au” audio files, “AVI” audio/video files, and MIDI files, may be streamed without requiring the copying or reformatting of the base media data. Since the base media data is not written to, but merely augmented by QuickTime declarations and hint information in separate files, the base media data may also be provided on read-only machine readable media such as CDROM.

In one embodiment of the present invention, the hint tracks embody the results of off-line computation and are typically optimized to provide the server with information to support packetization, and if needed, multiplexing.

10 Example hints, for example, for RTP (the IETF standard real-time protocol) and MPEG-2 transport are shown in Appendixes A-C.

In one embodiment of the present invention, a single file may support hint tracks for multiple protocols, or multiple different parameterizations of the same protocols, without undue space overhead. New protocols, and their associated hint tracks, may be designed without disrupting systems relying on existing protocols. Thus the invention, at least in one embodiment, is protocol-neutral.

In the QuickTime file format, a track may be added to the movie by updating or copying and augmenting the meta-data. If the media data is in files separate from the meta-data, or optimized interleave is not required, this can be a relatively simple and efficient operation.

In one embodiment of the present invention, tracks may be extracted by building a new set of movie meta-data which contains only one track, and which can, if desired, reference the media data in the original.

For example, in one embodiment of the present invention, a new audio track may be added which is marked as being an alternative to a set of other audio tracks. If it is also marked with the language code (e.g. French, or Tagalog), then the appropriate track may be selected at presentation time.

- 5 SMPTE time-code tracks are an example of elementary streams which may be present, added, or removed, as need arises, according to one embodiment of the invention.

- According to one aspect of the invention, hint tracks may permit the development of new formats for new protocols without causing compatibility issues
10 for existing servers or local playback. In addition, new media tracks may be added over the life of the file format while maintaining backwards compatibility.

In one embodiment of the present invention, the areas of extensibility include:

- 15 a) New track types which can be defined for media types not covered by the current QuickTime file format (e.g. laboratory instrument readings).
b) New coding types for existing tracks which may be defined (e.g. video or audio codecs). There is explicit provision for their codec-specific initialization information.
20 c) New hint track types which may be defined for new protocols, and a file which may contain hint information for more than one protocol without incurring a space overhead for the media data itself.

- Existing content on read-only media may be used with the present invention
25 (e.g., prepackaged movies on CD ROM, DVD, etc.).

Furthermore, according to one aspect of the invention, various "foreign" file formats may be used. In one embodiment of the present invention, for example, if the existing content is either in QuickTime format, or can be imported, it may be edited and streamed without requiring copying or re-formatting.

In one embodiment of the present invention, if a codec supports striping of the media data to achieve scalability of bandwidths, then these striped bandwidths may be represented using multiple stream tracks. Each track may represent a different bandwidth. Tracks may be grouped together in selected subsets of the basic media.

- 5 In one embodiment of the present invention, if a protocol supports bandwidth scalability, then the hint track itself may contain information for each protocol data unit (sample in the hint track). Information may include the bandwidth threshold above which the protocol data unit should be delivered to the network. Thus, hint tracks may indicate an available bandwidth as being high, low, etc., and/or other information
- 10 relating to bandwidth for data transmission.

In one embodiment of the present invention, if the protocol is a multiplexing protocol (e.g. MPEG-2 transport) then different hint tracks may be built which use a different subset of the elementary stream tracks to achieve different data-rates. Hence, some tracks may be omitted entirely for low bit-rate transmission.

- 15 In one embodiment of the present invention, if it is desired to record the base data using different codecs, then those tracks may be formed into a group of alternatives, and only one selected for presentation. The selection of which track to use for presentation is typically protocol-dependent and may be achieved by using the hint track approaches described herein.

- 20 In one embodiment of the present invention, encryption may also be pre-applied to a media file. In this case, the encrypted data may be stored in either (a) a new elementary stream (a new track) which is linked to the original media data (or the original media data may be removed if it is no longer needed) or (b) the hint track

itself. In case (b), it is possible that the hint track does not extract any data from the elementary un-encrypted stream on the fly. Thus, all of the media data may be in the hint track as well as the streaming packet protocol data unit information, because the media data may be transformed by encryption.

5 As an example of embedded object content information, the IETF session description information for a whole movie, and for individual tracks, may be stored in the meta-data for the RTP hint tracks, as user atoms.

10 In one embodiment of the present invention, a file format typically contains both media data in a playable format, and streaming information. In one embodiment, it is possible to stream directly from this format with relatively low overhead, while preserving the media independence, protocol independence, and ability to present the media locally.

15 According to one aspect of the invention, hint tracks may abstract detailed knowledge of codecs, timing and packetization, into an off-line preparation process. Thus, following the hint tracks to generate the data stream may be relatively simple and require no specialized knowledge of the media being streamed. Thus, decoupling of a server, for example, from the details of the data content may be provided, according to one aspect of the invention.

20 In one embodiment of the present invention, a set of hint tracks may be used to construct a file which is directly optimized for streaming—for example, by laying out network PDUs on disk at logical disk boundaries, in the time sequence in which they should sent. Such a file may no longer be a general presentation, but may be

streamed. In one embodiment, packetized files created with hint tracks may be stored and, for example, later optimized for streaming.

In one embodiment of the present invention, by encapsulating foreign file formats, media data may be retained in other formats while still be published in
5 QuickTime. For example, an existing format may be directly encapsulated into a new media data file by applying the proper wrapper, or may be left intact and referred to in segments or as a whole by the hint track, allowing the legacy formats to be streamed without copying. A single movie may contain pieces selected from multiple legacy formats. This invention does *not* constrain the base media format.

10 In general, a common format which spans capture, authoring and editing, download and streaming, will generally provide flexibility. Material may be reworked after use, or used in multiple ways, without being copied or re-formatted. In one embodiment of the present invention, it is possible to re-work and re-use material which has been hinted, by stripping the hint tracks, using standard editors, and then
15 re-hinting after editing is completed.

If it is desired that a media file be downloaded for local viewing, an optimized interleaved file may be built for that purpose, with the streaming meta-data in a separate declaration file referencing the same base media data. The download may not, therefore, include the streaming information, and yet the media data may be
20 present only once at a streaming server.

By separating logical structure from physical structure, the physical structure of the file may be optimized differently depending on the application (e.g. editing, local viewing, streaming).

By permitting the existence of multiple hint tracks for each media track, in one embodiment of the present invention, the file may be published by streaming over multiple protocols, without requiring multiple copies of the media.

Figure 6 is a diagram of a network of computer systems in which media data
5 may be processed, according to one embodiment of the present invention. As shown
in Figure 6, a number of client computer systems, one or more of which may
represent one implementation of the receiving system described above with reference
to Figure 3, are coupled together through an Internet 622. It will be appreciated that
the term "Internet" refers to a network of networks. Such networks may use a variety
10 of protocols for exchange of information, such as TCP/IP, ATM, SNA, SDI, etc.
The physical connections of the Internet and the protocols and communication
procedures of the Internet are well known to those in the art. Access to the Internet
103 is typically provided by Internet service providers (ISPs), such as the ISP 624
and the ISP 626. Users on client systems, such as the client computer systems 602,
15 604, 618, and 620, generally obtain access to the Internet through Internet service
providers, such as ISPs 624 and 626. Access to the Internet may facilitate transfer of
information (e.g., email, text files, media files, etc.) between two or more digital
processing systems, such as the client computer systems 602, 604, 618, and 620
and/or a Web server system 628. For example, one or more of the client computer
20 systems 602, 604, 618, and 620 and/or the Web server 628 may provide media data
(e.g., video and audio, or video, or audio) to another one or more of the client
computer systems 602, 604, 618, and 620 and/or the Web server 628. Such may be
provided in response to a request. As described herein, such media data may be

transferred in the system 600 according hints. Such hints, in one embodiment of the invention, may be created according to a specific format of the media data and/or a specific data communication (e.g., network) protocol(s).

The Web server 628 is typically comprised of at least one computer system to
5 operate with one or more data communication protocols, such as the protocols of the World Wide Web, and as such, is typically coupled to the Internet 622. Optionally, the Web server 628 may be part of an ISP which may provide access to the Internet and/or other network for client computer systems. The client computer systems 602, 604, 618, and 620 may each, with appropriate web browsing software, access data,
10 such as HTML documents (e.g., Web pages), which may be provided by the Web server 628. Such data may provide media, such as QuickTime movies, which may be presented by the client computer systems 602, 604, 618, and 620.

The ISP 624 provides Internet connectivity to the client computer system 602 via a modem interface 606, which may be considered as part of the client computer
15 system 602. The client computer system may be a conventional computer system, such as a Macintosh computer, a "network" computer, a handheld/portable computer, a Web TV system, or other types of digital processing systems (e.g., a cellular telephone having digital processing capabilities). Similarly, the ISP 626 provides Internet connectivity for the client computer systems 604, 618 and 620, although as
20 depicted in Figure 6, such connectivity may vary between various client computer systems, such as the client computer systems 602, 604, 618, and 620. For example, as shown in Figure 6, the client computer system 604 is coupled to the ISP 626 through a modem interface 608, while the client computer systems 618 and 620 are

part of a Local Area Network (LAN). The interfaces 606 and 608, shown as modems 606 and 608, respectively, in Figure 6, may be an analog modem, an ISDN modem, a cable modem, a satellite transmission interface (e.g., "Direct PC"), a wireless interface, or other interface for coupling a digital processing system, such as a client computer system, to another digital processing system. The client computer systems 618 and 620 are coupled to a LAN bus 612 through network interfaces 614 and 616, respectively. The network interfaces 614 and 616 may be an Ethernet-type, Asynchronous Transfer Mode (ATM), or other type of network interface. The LAN bus is also coupled to a gateway digital processing system 610, which may provide firewall and other Internet-related services for a LAN. The gateway digital processing system 610, in turn, is coupled to the ISP 626 to provide Internet connectivity to the client computer systems 618 and 620. The gateway digital processing system 610 may, for example, include a conventional server computer system. Similarly, the Web server 628 may, for example, include a conventional server computer system.

15 The system 600 may allow one or more of the client computer systems 602, 604, 618, and 620 and/or the Web server 628 to provide media data (e.g., video and audio, or video, or audio) to another one or more of the client computer systems 602, 604, 618, and 620 and/or the Web server 628. Such data may be provided, for example, in response to a request by a receiving system, which may be, for example,

20 one or more of the client computer systems 602, 604, 618, and 620. As described herein, such media data may be transferred in the system 600 according hints or hint tracks. Such hints, in one embodiment of the invention, may be created according to a specific format of the media data and/or a specific data communication (e.g., network)

protocol(s) to allow, according to one aspect of the invention, packetization of media data.

Figure 7 is a block diagram of a digital processing system which may be used in accordance with one embodiment of the present invention. For example, the digital processing system 650 shown in Figure 7 may be used as a client computer system, a Web server system, a conventional server system, etc. Furthermore, the digital processing system 650 may be used to perform one or more functions of an Internet service provider, such as the ISP 624 or 626. The digital processing system 650 may be interfaced to external systems through a modem or network interface 668. It will be appreciated that the modem or network interface 668 may be considered as part of the digital processing system 650. The modem or network interface 668 may be an analog modem, an ISDN modem, a cable modem, a token ring interface, a satellite transmission interface, a wireless interface, or other interface(s) for providing a data communication link between two or more digital processing systems.

The digital processing system 650 includes a processor 652, which may represent one or more processors and may include one or more conventional types of such processors, such as a Motorola PowerPC processor, an Intel Pentium (or x86) processor, etc. A memory 155 is coupled to the processor 652 by a bus 656. The memory 155 may be a dynamic random access memory (DRAM) and/or may include static RAM (SRAM). The processor may also be coupled to other types of storage areas/memories (e.g., cache, Flash memory, disk, etc.), which could be considered as part of the memory 155 or separate from the memory 155.

The bus 656 further couples the processor 652 to a display controller 658, a mass memory 662, the modem or network interface 668, and an input/output (I/O) controller 664. The mass memory 662 may represent a magnetic, optical, magneto-optical, tape, and/or other type of machine-readable medium/device for storing
5 information. For example, the mass memory 662 may represent a hard disk, a read-only or writeable optical CD, etc. The display controller 658 controls in a conventional manner a display 660, which may represent a cathode ray tube (CRT) display, a liquid crystal display (LCD), a plasma display, or other type of display device. The I/O controller 664 controls I/O device(s) 666, which may include one or
10 more keyboards, mouse/trackball or other pointing devices, magnetic and/or optical disk drives, printers, scanners, digital cameras, microphones, etc.

It will be appreciated that the digital processing system 650 represents only one example of a system, which may have many different configurations and architectures, and which may be employed with the present invention. For example,
15 Macintosh and Intel systems often have multiple busses, such as a peripheral bus, a dedicated cache bus, etc. On the other hand, a network computer, which may be used as a digital processing device of the present invention, may not include, for example, a hard disk or other mass storage device, but may receive routines and/or data from a network connection, such as the modem or interface 668, to be processed by the
20 processor 652. Similarly, a Web TV system, which is known in the art, may be considered to be a digital processing system of the present invention, but such a system may not include one or more I/O devices, such as those described above with reference to I/O device(s) 666. Additionally, a portable communication and data

processing system, which may employ a cellular telephone and/or paging capabilities, may be considered a digital processing system which may be used with the present invention.

In the system 650 shown in Figure 7, the mass memory 662 (and/or the
5 memory 654) may store media (e.g., video, audio, movies, etc.) which may be processed according the present invention (e.g., by way of hints). Alternatively, media data may be received by the digital processing system 650, for example, via the modem or network interface 668, and stored and/or presented by the display 660 and/or I/O device(s) 666. In one embodiment, packetized media data may be
10 transmitted across a data communication network, such as a LAN and/or the Internet, in accordance with hint tracks. On the other hand, the processor 652 may execute one or more routines to use a file with one or more hint tracks, or alternatively, to create one or more hint tracks, to process media (e.g., a pre-packaged movie, audio file, video file, etc.) for presentation or packetization according to the hint tracks. Such
15 routines may be stored in the mass memory 662, the memory 664, and/or another machine-readable medium accessible by the digital processing system 650. In one embodiment, the digital processing system 650 may process media data having hint tracks embedded therein. Similarly, such embedded media data may be stored in the mass memory 662, the memory 664, and/or another machine-readable medium
20 accessible by the digital processing system 650.

Figure 8 is a block diagram of a system that utilizes hints to transfer media data, according to one embodiment of the invention. The system 680 shown in Figure 8 includes a receiving system, which is depicted as a client data processing system

682 coupled to a server 694, via a data communication link 686. The server 694 and/or client data processing system may, for example, represent one or a combination of the devices/systems described with reference to Figures 6 and 7.

5 The server 694 includes a hint generation and processing unit 688, a media processing unit 690, and a data communication unit 692, each of which may include hard-wired circuitry or machine-executable instructions or a combination thereof. Furthermore, at least a portion of such hard-wired circuitry and/or machine-executable instructions may be shared between a combination of the hint generation and processing unit 688, the media processing unit 690, and the data communication unit 10 692. In one embodiment, at least one storage area/memory (e.g., a machine-readable medium) having appropriate routines and/or data stored therein coupled to at least one processor is utilized, at least in part, to implement one or a combination of the hint generation and processing unit 688, the media processing unit 690, and the data communication unit 692.

15 In one embodiment, the hint generation and processing unit 688 creates and stores hints for packetization of media data processed by the media processing unit 690. As described above, the hints may be generated and stored as a separate file, relative to media files or may be embedded with media data. If more than one media format is to be processed, an appropriate format may be taken into consideration by 20 the hint generation and processing unit 688 to generate the hints. Information about the media format may be provided by the media processing unit 690, which may also provide the media data (e.g., media files of video, audio, or video and audio, etc.). Similarly, the data communication unit 692 may provide one or more data

communication (e.g., network) protocols for exchange of such media data, packetized according to the hints, via the data communication link 686. As such, the hint generation and processing unit may determine, based on media format information provided by the media processing unit 690 and data communication protocol information provided by the data communication unit 692, appropriate hints and packetization of media and/or the hints for transfer to a receiving digital processing system, such as the client data processing system 682. In one embodiment, the streaming of the media and hints is done in accordance with the QuickTime format.

In response to media data and hint packets received via the data communication link 686, the client data processing system 682 may present a media object represented by the media data. Such presentation may be performed ephemerally, as described above. In one embodiment of the invention, the media data may optionally be stored by the client data processing system 682 and reassembled, for example, at a later time, for presentation and/or transmission by the client data processing system 682.

Figure 9 is a block diagram of a system that utilizes hints to transfer media data, according to one embodiment of the invention. In particular, Figure 9 depicts an embodiment of the invention wherein a separate digital processing system, referred to as a generator, may generate hints (or hint tracks) to provide to another system, such as a server, that uses the hints to packetize media data for transfer to another system, such as a client computer system. A system 696 is shown in Figure 9, which includes a server 700 which may exchange data, via the data communication link 686, with the client data processing system 682. However, in the embodiment shown in Figure 9, the server 700 does not generate the hints. Rather, a generator 710, coupled to the

server 700 by a data communication link 708, includes a hint generation unit 712 to generate hints that are used to packetize media data.

In one embodiment, the operation of the system 696 is as follows: the server 700 makes a request to the generator 710 to generate hints for one or more media files
5 containing media data. For example, the media files may be stored in the server 700 on a machine-readable medium. The request may include information to indicate the format of the media file and/or a data communication protocol for transmission of the media data and/or other data. The data communication protocol may be related to the data communication link 686, which may, in one embodiment of the invention, be
10 associated with a network connection having particular physical and logical characteristics to facilitate exchange of media and/or other data between the server 700 and the client data processing system 682. In response to the request, the hint generation unit 712 generates appropriate hints, which may be associated with a time-related hint track, and provides the hints to the server 700. In response to the hints
15 received from the generator 710, via the data communication link 708, the server 700, and in particular, a hint processing unit 702 uses the hints to packetize the media data for transmission to the client data processing system 682.

In response to media data and hint packets received via the data communication link 686, the client data processing system 682 may present a media object represented
20 by the media data. Such presentation may be performed ephemerally, as described above. In one embodiment of the invention, the media data may optionally be stored by the client data processing system 682 and reassembled, for example, at a later time, for presentation and/or transmission by the client data processing system 682.

Figure 10 is a flow diagram illustrating a method for generating hints for providing media data transmission, according to one embodiment of the invention. In step 720, a media format is determined for media data to be transmitted, if more than one format will be used. If only one format is used, 720 may not be performed. In step 722, an appropriate data communication protocol(s) is determined, again, assuming that more than one (protocol) may be used. In step 724, based on the media format and the data communication protocol(s) (one or both of which may have been selected/configured), hints (e.g., hint tracks) related to media data transmission are created and stored.

10 In step 726, which is optional, the hints may be transmitted to another digital processing system. In one embodiment of the invention, for example, the method of Figure 10, at least in part, may be performed exclusively by one digital processing system (e.g., a server). In an alternative embodiment, the method of Figure 10, at least in part, may be performed by two or more digital processing systems. For
15 example, attributes of media data may be provided by a server or other system to another digital processing system, such as a generator. In response, the generator may determine, based on the attributes, an appropriate media format, data communication protocol(s), and hints for packetization of media data, which may be stored at the server. Alternatively, the server may provide the appropriate media
20 format and protocol(s) to the generator, which could then generate hints. The generator may transmit the hints to the server or other digital processing system, which could packetize media data according to the hints.

Figure 11 is a flow diagram illustrating a method of processing media data received by a receiving system in accordance with hints, according to one embodiment of the invention. In step 730, media data transmitted according to a receiving system in accordance with hints or hint tracks is received by the receiving system. In one
5 embodiment, the receiving system may receive packetized media data, as well as packetized hint tracks. The hint tracks, in one embodiment of the invention, may be associated with at least portions of the media data. Such data may be received by the receiving system in response to a request that may be made by the receiving system. For example, in one embodiment, the receiving system may be a client computer
10 system and the request may be made to a server or other digital processing system for the media data. In response, the server may generate (or have generated for it by a separate digital processing system) hints for packetizing the media data, and transmit the packetized media data, which may include hints, to the receiving system.

In step 732, a media object represented by the media data received by the
15 receiving system is presented by the receiving system. For example, the media data may include video, audio, or combination thereof that is "presented" by the receiving system, for example, on a display and speaker(s). As mentioned above, the media data may be associated with a QuickTime movie.

Optionally, in step 734, the media data, which may include hints, may be
20 stored by the receiving system as a media file(s). Thus, in alternative embodiments of the invention, step 732 may not be performed as the media data is received, or may be performed before, after, or in parallel with step 734.

In step 734, the stored media file may optionally be reassembled and/or presented. As such, step 732 may be performed subsequent to step 734.

Figure 12 is an example of a machine readable storage medium that may be accessed by a digital processing system, such as a generator, according to one
5 embodiment of the invention. It will be appreciated that the actual memory that stores the elements shown in and described below with reference to Figure 12 may be one or several elements, such as one or more disks (which may, for example, be magnetic, optical, magneto-optical, etc.), the memory 654 and/or the mass memory 662 described above with reference to Figure 7. Furthermore, in one embodiment where
10 the generator, with which the machine readable storage medium shown in Figure 12 is associated, is a network computer, one or more of the elements of the machine readable storage medium may be stored at another digital processing system and downloaded to the generator. Furthermore, the elements described with reference to the machine readable storage medium may, at some point in time, be stored in a non-
15 volatile mass memory (e.g., a hard disk). Conversely, at other times, the elements of the machine storage medium may be dispersed between different storage areas, such as DRAM, SRAM, disk, etc.

Figure 12 shows a machine readable storage medium 740. In one embodiment, the machine readable storage medium is utilized, at least in part, by a
20 digital processing system that generates hints or hint tracks, i.e., a generator, in accordance with one or more method(s) of the invention. The generator, as described with reference to Figure 8, may be integrated into a digital processing system that transmits media data according to the hint tracks, or may be, as described with

reference to Figure 9, a digital processing system that creates and provides the hints to another digital processing system, such as a server, which utilizes the hints to packetize and transmit media data.

As shown in Figure 12, the machine readable storage medium 740 typically
5 includes a number of elements. For example, the machine readable storage medium 740 includes software for providing operating system functionality to the generator, as depicted by a generator operating system (OS) 742. A network transmission routine(s) 748 provides data communication functionality, such as routines, protocols, etc., to allow the generator to transmit and receive data via a data communication link.

10 In addition, the machine readable storage medium 740 includes routines and data for creating hints associated with media transmission. As such, the machine readable storage medium 740 may optionally include information 750, which may provide information relating to one or more data communication protocols and media formats which may be necessary for creation of hints by a hint creation routine(s) 744.
15 For example, the information 750 may include information relating to QuickTime movies, RTP, MPEG, etc. However, such information may, at least in part, be integrated into the hint creation routine 744 and/or be provided to the generator by a remote digital processing system.

The hints created by the hint creation routine(s) 744 may be stored as created
20 hints 746 and/or stored/transmitted elsewhere (e.g., at a remote digital processing device, which may be a server). The hints are hint tracks that are time-related for packetization and transmission of media data, which is also time-related (e.g., video, audio, video and audio, etc.).

Although the machine readable storage medium 740 is described with reference to a generator, the medium 740, at least in part, may be part of a number of types of digital processing systems, data storage media, etc. For example, the machine readable storage medium 740, at least in part, may be included as part of a server or other digital processing system. Furthermore, the machine readable storage medium 740, at least in part, may be included as part of a software utility on one or more disks or other machine readable media.

Figure 13 is an example of a machine readable storage medium that may be accessed by a digital processing system, such as a server, according to one embodiment of the invention. It will be appreciated that the actual memory that stores the elements shown in and described below with reference to Figure 13 may be one or several elements, such as one or more disks (which may, for example be magnetic, optical, magneto-optical, etc.), the memory 654 and/or the mass memory 662 described above with reference to Figure 7. Furthermore, in one embodiment where the server, with which the machine readable storage medium shown in Figure 13 is associated, is a network computer, one or more of the elements of the machine readable storage medium may be stored at another digital processing system and downloaded to the server. Furthermore, the elements described with reference to the machine readable storage medium may, at some point in time, be stored in a non-volatile mass memory (e.g., a hard disk). Conversely, at other times, the elements of the machine storage medium may be dispersed between different storage areas, such as DRAM, SRAM, disk, etc.

Figure 13 shows a machine readable storage medium 760. In one embodiment, the machine readable storage medium is utilized, at least in part, to packetize media data for transmission on a data communication link in accordance with one or more method(s) of the invention. The machine readable storage medium 760

5 may be associated with a server, such as the server 694 described with reference to Figure 8, to include routines to create hint tracks and transmit media data according to the hint tracks. In another embodiment, the machine readable storage medium 760 may be associated with a digital processing system, such as the server 700 described with reference to Figure 9, wherein a digital processing system, such a generator,

10 includes routines to create hints, and the server, using the hints as processed by routines provided by the machine readable storage medium 760, may packetize and transmit media data.

The machine readable storage medium 760 includes a number of elements. For example, the machine readable storage medium 760 includes software for

15 providing operating system functionality to the server, as depicted by a server operating system (OS) 762. A network transmission routine(s) 768 provides data communication functionality, such as routines, protocols, etc., to allow the server to transmit and receive data via a data communication link.

In addition, the machine readable storage medium 760 includes a media

20 packetization routine 770 for packetizing media data, which may be time-related, based on hints, and which may also be packetized. Accordingly, the machine readable storage medium 760 includes a media data storage area 764 and a hint storage area 766 to store media data (which may, for example, be QuickTime movies or other media

tracks) and hints (e.g., hint tracks), respectively. The hints may include hint tracks that are time-related for packetization and transmission of media data, which is also typically time-related (e.g., video, audio, video and audio). In one embodiment, the hint tracks are packetized separately from the media data packets. In one embodiment, 5 hints include pointer information identifying media data (e.g., a particular packet(s)) which may be in a separate media file.

Figure 14 is an example of a machine readable storage medium that may be accessed by a digital processing system, such as a receiving system or other digital processing system, according to one embodiment of the invention. It will be 10 appreciated that the actual memory that stores the elements shown in and described below with reference to Figure 14 may be one or several elements, such as one or more disks (which may, for example be magnetic, optical, magneto-optical, etc.), the memory 654 and/or the mass memory 662 described above with reference to Figure 7. Furthermore, in one embodiment where the receiving system, with which the machine 15 readable storage medium shown in Figure 14 is associated, is a network computer, one or more of the elements of the machine readable storage medium may be stored at another digital processing system and downloaded to the receiving system. Furthermore, the elements described with reference to the machine readable storage medium may, at some point in time, be stored in a non-volatile mass memory (e.g., a 20 hard disk). Conversely, at other times, the elements of the machine storage medium may be dispersed between different storage areas, such as DRAM, SRAM, disk, etc.

Figure 14 shows a machine readable storage medium 780. In one embodiment, the machine readable storage medium is utilized, at least in part, to

process media data packetized in accordance with one or more method(s) of the invention. The machine readable storage medium 780 may be associated with a receiving system, such as the client data processing system 682 described with reference to Figures 8 and 9, to include routines to present media data

5 transmitted/received according to hints. Alternatively, the machine readable storage medium 780 may include media data having hints (e.g., hint tracks) embedded therein. Such embedded media data may be pre-packaged or generated by a routine stored on a machine readable storage medium, such as the machine readable storage medium 780.

10 The machine readable storage medium 780 may include a number of elements. For example, the machine readable storage medium 780 includes software for providing operating system functionality to the receiving system, as depicted by a server operating system (OS) 772. A network transmission routine(s) 782 provides data communication functionality, such as routines, protocols, etc., to allow the server
15 to transmit and receive data via a data communication link.

In addition, the machine readable storage medium 780 includes a media presentation routine 778 for presenting media data packetized according to hints. Thus, the machine readable storage medium 780, and in particular, the media presentation routine 778, may include routines for decompression of audio and/or
20 video data, displaying of video, and/or playing back audio, etc. Furthermore, the media presentation routine 778 typically provides handling of hints that are associated with the media data. In one embodiment, the hints are simply ignored as media is presented.

Optionally, the machine readable storage medium 780 may store media data that has been packetized according to hints as media data 774, and include a media data reassembly routine 776 to reassemble to the stored media data (e.g., to be presented, transmitted, etc.).

5 Figure 15 is a diagram of a data storage and/or communication medium having stored/transported thereon media and hint information, according to one embodiment of the invention. A data storage and/or communication medium (medium) 800 is shown, which represents various types of transport and/or storage medium in which a media data packet 804 and a hint packet 806 packetized according to the present
10 invention could be stored or transported. For example, the medium 800 may represent the mass memory 662 and/or the memory 654, described above with reference to Figure 7. The medium 800 may also represent a communication medium, such as the LAN bus 612 shown in Figure 6 or the data communication link 686 for transporting data/signals representing media and/or other information.

15 The hint packet 806 and the media packet 804 may be integrated into one packet or be stored and/or transported separately, as depicted in Figure 15. Furthermore, the hint packet 806 and the media packet 804 may embody several types of formats, such as ones described herein or one associated with other media formats, network protocols, and/or digital processing device architecture.

20 Provided below are some example formats of hints. It will be appreciated that the present invention, however, may be utilized with various types of network protocols, digital processing system architectures, media formats, etc., to provide transmission of time-based data.

ALTERNATIVE EMBODIMENTS

While the invention has been described in terms of several embodiments and illustrative figures, those skilled in the art will recognize that the invention is not limited to the embodiments or figures described. In particular, the invention can be practiced in several alternative embodiments that provide packetization of time related media data.

Therefore, it should be understood that the method and apparatus of the invention can be practiced with modification and alteration within the spirit and scope of the appended claims. The description is thus to be regarded as illustrative instead of limiting on the invention.

Appendix A - Packetization Hint Sample Description

In one embodiment of the present invention, each hint track has a table of sample descriptions. Hint tracks typically have one sample description. The format for each sample description entry for a hint track, according to one embodiment of the present invention, is described below in Table 1.

Table 1: Hint Track Sample Description Format

Hint Track Sample Description	Bytes
Sample description size	4
Data format	4
Reserved	6
Data reference index	2
Max packet size	4
Additional data table	variable

The packetization hint header atom contains the following data elements:

Field descriptions:

5	Sample description size	A 32-bit integer that specifies the number of bytes in the sample description.
10	Data format	A 32-bit integer indicating the format of the hints stored in the sample data. Different formats may be defined for different hint types. The table below lists defined formats.
15	Reserved	Six bytes that are set to 0.
20	Data reference	A 16-bit integer that contains the index of the data index associated with the samples that use this sample description. Data references are stored in data reference atoms.
25	Max packet size	A 32-bit integer indicating the maximum size of packets computed in this track.
30	Additional Data Table	A table containing additional information needed on a per track basis. The values are tagged entries. There are no required entries. If an entry is not present in the table, a reasonable default may be used.

The structure for the additional data table entries is shown in Table 2.

Table 2: Additional Data Table Format

Additional Data Table	Bytes
Entry length	4
Data type	4
Data	Entry length - 8

The additional data table entries contain the following data elements:

Field descriptions:

5 Entry length A 32-bit integer indicating the length of the entire entry (includes 8 bytes for the length and type fields) in bytes.

 Data type A 32-bit integer indicating the meaning of the data in the entry.

10 Data The data for this entry. The length of the data is indicated by the Data length field of the table.

The following data tags may defined for several various types of data format types. Other tags may be created as required.

Length	Type	Data Description
9	'rely'	A 1 byte integer indicating whether or not this track should be sent over a reliable transport. Values of 0 and 1 are defined. If this tag is not present, it is assumed to have the value zero, indicating that it can be sent over unreliable transports, such as UDP.

15

20

The following data format types are defined. New types may be defined as needed.

Data Format	Description
'rtp'	The packetization hints for sending media over RTP for the specific media type and encoding as described by various IETF drafts of the Audio-Video Transport (AVT) working group.

5

The following data tag is utilized in one embodiment for 'rtp' data.

Length	Type	Data Description
12	'tims'	A 32-bit number indicating the RTP timescale. This tag is present in one embodiment for RTP data.

10

The following data tags are optional for 'rtp' data.

Length	Type	Data Description
12	'tsro'	A 32-bit number indicating the random offset to add to the stored time stamp when sending the RTP packets. If this field is not present, a truly random number should be used, as per the RTP specification. The value of this field could be zero, indicating that no random offset is to be added.
10	'snro'	A 16-bit number indicating the random offset to add to the sequence

15

20

5

[illegible]

10

15

20

The hint track is related to its base media track by a single track reference declaration. (RTP does not permit multiplexing of media within a single RTP stream). The sample description for RTP declares the maximum packet size which this hint track will generate. Session description (SAP/SDP) information is stored in user-data atoms in the track.

Each sample in the RTP hint track contains the instructions to send out a set of packets which must be emitted at a given time. The time in the hint track is emission time, not necessarily the media time of the associated media.

In the following description the internal structure of samples, which are media data, not meta data, in the terminology of this example is described, need not be structured as objects.

In this example, each sample contains two areas: the instructions to compose the packets, and any extra data needed when sending those packets (e.g. an encrypted version of the media data).

```
15 struct RTPsample {  
    int(16)    packetcount;  
    RTPpacket  packets[packetcount];  
    byte[]     extradata;  
}
```

20 Each RTP hint packet contains the information to send a single packet. In one embodiment, to separate media time from emission time, an RTP time stamp is specifically included, along with data needed to form the RTP header. In alternative embodiments, however, this is not the case. Other header information is typically supplied. A table of construction entries is constructed as follows:

```
struct RTPpacket {  
    int(32)    RTPtime;  
    int(16)    partialRTPheader;  
    int(16)    RTPsequenceseed;  
5    int(16)    entrycount;  
    dataentry  constructors[entrycount];  
}
```

There are various forms of the constructor. Each constructor is 16 bytes, which may make iteration relatively simple. The first byte is a union discriminator:

```
10 struct dataentry {  
    int(8) entrytype;  
    switch entrytype {  
        case immediate:  
            int(8)    bytecount;  
15            int(8)    bytestocopy[bytecount];  
        case mediasample:  
            int(8)    reserved[5];  
            int(16)   length;  
            int(32)   mediasamplenumber;  
20            int(32)   mediasampleoffset;  
        case hintsample:  
            int(8)    reserved[5];  
            int(16)   length;  
            int(32)   hintsamplenumber;  
25            int(32)   hintsampleoffset;  
    }  
}
```

The immediate mode permits the insertion of payload-specific headers (e.g. the RTP H.261 header). For hint tracks where the media is sent 'in the clear', the mediasample entry may specify the bytes to copy from the media track, by giving the sample number, data offset, and length to copy. For relatively complex cases (e.g. encryption or forward error correction), the transformed data may be placed into the hint samples, and then hintsample mode may be used, which would be provided from the extradata field in the RTPsample itself.

In one example of the present invention, there is no requirement that successive packets transmit successive bytes from the media stream. For example, to

conform with RTP-standard packing of H.261, in one example of the present invention, a byte may be sent at the end of one packet and also at the beginning of the next (when a macroblock boundary falls within a byte).

5 **Appendix C - Packetization Hint Sample Data for Data Format 'rtp'**

 This appendix provides a description of the sample data for the 'rtp' format, according to one embodiment of the invention. The 'rtp' format assumes that a server is sending data using Real Time Transport Protocol (RTP). This format assumes that the server knows about RTP headers, but does not require that the server know
10 anything about specific media header, including media headers defined in various IETF drafts.

 In one embodiment of the present invention, each sample in the hint track will generate one or more RTP packets. Each entry in the sample data table in a hint track sample corresponds to a single RTP packet. Samples in the hint track may or may not
15 correspond exactly to samples in the media track. In one embodiment of the present invention, data in the hint track sample is byte aligned, but not 32-bit aligned.

Field descriptions:

Entry count	A 16-bit unsigned integer indicating the number of packet entries in the table. Each entry in the table corresponds to a packet. Multiple entries in a single sample indicate that the media sample had to be split
-------------	---

20

into multiple packets. A sample with an entry count of zero is reserved and if encountered, should be skipped.

Packet entry table A variable length table containing packet entries.

Packet entries are defined below.

- 5 Additional data A variable length field containing data pointed to by the entries in the data table shown below by Table 3:

Table 3 - Additional Data

Packet Entry	Bytes
Relative packet transmission time	4
Flags	4
RTP header info	2
RTP sequence number	2
Entry count	2
Data table	variable

In one embodiment, the packet entry contains the following data elements:

10 Field descriptions:

relative packet
transmission time

A 32-bit signed integer value, indicating the time,
in hint track's timescale, to send this packet relative
to the hint sample's actual time. Negative values
mean that the packet will be sent earlier than real

15

time, which is useful for smoothing the data rate. Positive values are useful for repeating packets at later times. Within each hint sample track, each packet time stamp is nondecreasing.

5 flags A 32-bit field indicating certain attributes for this packet.

The RTP header information field contains the following element:

	Field	Bit #	Description
10	R	31	A 1-bit number indicating that this is a repeat packet - the data has been defined in a previous packet. A server may choose to skip repeat packets to help it catch up when it is behind in its transmission of packets. All repeated packets for a given packet care in the same hint sample.
15			All undefined bits (0-30) are reserved and are set to zero.
20	RTP header info		A 16-bit integer specifying various values to be set in the RTP header.

The RTP header information field contains the following elements:

Field	Bit#	Description
-------	------	-------------

5	P	2	A 1-bit number corresponding to the padding (P) bit in the RTP header. This bit may not be set, since a server that needed different packet padding may generally need to un-pad and re-pad the packet itself.
10	X	3	A 1-bit number corresponding to the extension (X) bit in the RTP header. This bit may not be set, since a server that needs to send its own RTP extension may either not be able to, or may be forced to replace any extensions from the hint track.
	M	8	A 1-bit number corresponding to the marker (M) bit in the RTP header.
15	payload type	9-15	A 7-bit number corresponding to the payload type (PT) field of the RTP header.

20 All undefined bits (0-1 and 4-7) are reserved and are set to zero. The location of the defined bits are in the same bit location as in the RTP header.

25 RTP sequence number A 16-bit integer specifying the RTP sequence number for the packet. The RTP server adds a random offset to this sequence number before transmitting the packet. This field allows re-transmission of packets, e.g., the same packet can

be assembled with the same sequence number and a different (later) packet transmission time. For example, a text sample with a duration of 5 minutes can be retransmitted every 10 seconds so that clients that miss the original sample transmission (perhaps they started playing a movie in the middle) will be "refreshed" after a maximum of 10 seconds.

5

Entry count A 16-bit unsigned integer specifying the number of entries in the data table.

Data table A table that defines the data to be put in the payload portion of the RTP packet. This table defines various places the data can be retrieved, and is shown by Table 4.

10

Table 4 - Data Table

Data table entry	Bytes
Data source	1
Data	15

The data source field of the entry table indicates how the other 15 bytes of the entry are to be interpreted. Values of 0 through 4 are defined. The various data table formats are defined below. Although there are various schemes, the entries in the various schemes are typically 16 bytes long.

15

No-Op Data Mode

The data table entry has the following format for no-op mode:

Field description:

Data source = 0 A value of zero indicates that this data table entry is to be ignored.

5 Immediate Data Mode

The data table entry has the following format for immediate mode:

Field description:

	Data source = 1	A value of one indicates that the data is to be immediately taken from the bytes of data that follow.
10	Immediate length	An 8-bit integer indicating the number of bytes to take from the data that follows. Legal values range from 0 to 14.
15	Immediate data	14 bytes of data to place into the payload portion of the packet. Only the first number of bytes indicated by the immediate length field are used.

Sample Mode

The data table entry has the following format for sample mode:

Field description:

20	Data source =2	A value of two indicates that the data is to be taken from a track's sample data.
	Track ref index.	A value that indicates which track the sample data will come from. A value of zero means that there is exactly one media track referenced, which is to be used. Values from 1 to

127 are indices into the hint track reference atom entries, indicating from which original media track the sample is to be read. A value of -1 means the hint track itself, i.e., the sample from the same track as the hint sample currently being parsed is used.

5

Bytes per
compression
block

A 16-bit unsigned integer specifying the number of bytes that results from compressing the number of samples in the Samples per compression block field. A value of zero is equivalent to a value of 1.

10

Samples per
compression
block

A 16-bit unsigned integer specifying the uncompressed samples per compression block. A value of zero is equivalent to a value of 1.

15

Length

A 16-bit integer specifying the number of bytes in the sample to copy.

Sample Number

A 32-bit integer specifying sample number of the track.

Offset

A 32-bit integer specifying the offset from the start of the sample from which to start copying. If referencing samples in the hint track, this will generally point into the Additional Data area.

20

If the bytes per compression block and/or the samples per compression block is greater than 1, then this ratio is used to translate a sample number into an actual byte offset. This ratio mode is typically used for compressed audio tracks in QuickTime movies, such that:

5 $CB = NS * BPCB / SPCB$

wherein,

10 CB = compressed bytes
 NS = number of samples
 BPCB = bytes per compression block
 SPCB = samples per compression block

For example, a GSM compression block is typically 160 samples packed into 33 bytes. Therefore, BPCB = 33 and SPCB = 160. The hint sample requests 33 bytes of data starting at the 161st media sample. Assuming that the first QuickTime chunk contains at least 320 samples, so after determining that this data will come from chunk 1, and where chunk 1 starts, the ratio is utilized to adjust the offset into the file where the requested samples will be found:

20 chunk_number = 1; /* calculated by walking the sample-to-chunk atom */
 first_sample_in_this_chunk = 1; /* also calculated from that atom */
 chunk_offset = chunk_offsets[chunk_number]; /* from the stco atom */
 data_offset = (sample_number - first_sample_in_this_chunk) * BPP / SPP
 read_from_file(chunk_offset + data_offset, length); /* read our data */

Sample Description Mode

The data table entry has the following format for sample description mode:

Field description:

	Data source = 3	A value of three indicates that the data is to be taken from the media track's sample description table.
5	Track ref index	A value that indicates which track the sample data will come from. A value of zero means that there is exactly one hint track reference, which is to be used. Values from 1 to 127 are indices into the hint track reference atom entries, indicating from which original media track the sample is to be read. A value of -1 means the hint track itself, i.e., the sample description from the same track as the hint sample currently being parsed is utilized.
10	Reserved	Four bytes that are set to zero.
	Length	A 16-bit integer specifying the number of bytes in the sample to copy.
15	Sample description index	A 32-bit integer specifying the index into the media's sample description table.
	Offset	A 32-bit integer specifying the offset from the start of the sample from which to start copying.
20	Additional data	A variable length field containing data pointed to by hint track sample mode entries in the data table.

Appendix D — Example hint track format for MPEG-2 Transport

This section presents one example of a simple track format for streaming MPEG-2 transport from a QuickTime movie holding elementary streams.

5 An MPEG-2 transport stream is associated with a multiplex of one or more elementary streams. For this reason, an MPEG-2 transport hint track describes how to construct such a multiplex from one or more media tracks. There is not necessarily a one to one relationship between media tracks and MPEG-2 transport hint tracks. Each hint track may contain references to the elementary streams it represents. In one
10 example of the present invention, a QuickTime file might contain multiple such hint tracks to describe different multiplexes.

 Packet size is generally not an issue, since all MPEG-2 transport packets are 188 bytes in size. In one example of the present invention, each transport packet (in the MPEG-2 transport protocol) contains payload data from one media track. This
15 allows for a relatively simple hint description for each transport packet. In one example of the present invention, each such hint describes which header data appears on each transport packet, and then points to the payload in the appropriate media track for the transport packet. For packets which do not correspond with a media track, such as PSI packets, the hint may describe 188 bytes of header data, and any media
20 track reference may be considered irrelevant. For packets which do correspond with a media track, the header data may account for information such as transport headers, possible adaptation headers, and PES headers for transport packets that begin PES packets.

Reference is made to the MPEG-2 transport hint track in the Sample Description Atom (of type 'std'). This atom includes a sample description table, and the entries in this table differ based on the media type. In one example of the present invention, hint tracks begin with the structure shown in Table 1. The additional data
5 table may hold entries with the structure shown in Table 2:

In one example of the present invention, if the hint track is an MPEG-2 transport hint track, the data format in the hint track sample description entry will be 'm2t' and the max packet size will always be 188. In such a description entry, the types shown below in Tables 5-7 may be found in the additional data table:

Table 5 - Additional Data Table Entries

Entry length	Data type	Data description
8	0x00000000	Indicates there are no more entries in the table
9	'otyp'	Describes how offsets are described in the hints. The one byte of data has values described below in figure B.4. <i>This entry is mandatory in the additional data table.</i>
9	'msns'	Describes the size of media sample numbers. The one byte of data indicates how many bytes are used to specify media sample numbers. If this is not present, and media sample numbers are present in the sample data, the default value is 4 bytes.
9	'msos'	Describes the size of media sample offsets. The one byte of data indicates how many bytes are used to specify media sample offsets. If this is not present, and media sample offsets are present in the sample data, the default value is 4 bytes.
9	'fosz'	Describes the size of file offsets. The one byte of data indicates how many bytes are used to specify file offsets within samples. If this is not present, and file offsets are present in the sample data, the default value is 4 bytes.
Variable	'tmap'	Describes an abbreviated mapping of media tracks. Each 5 byte entry maps a 4 byte track ID to a 1 byte track reference number. This limits any given transport mux to containing no more than 256 media tracks, but this should not be a limiting factor, and this compression is useful in limiting the size of the hint track. The format of these 5 byte entries is specified below in figure B.5. <i>This entry is mandatory in the additional data table.</i>

Table 6- 'otyp' Values In the Additional Data Table

Value	Description
0	Samples are described in terms of media samples
1	Samples are described in terms of file offsets

Table 7 - Format of Entries in the 'tmap' Additional Data Entry

Length	Description
4	Original Track ID
1	Abbreviated track reference number used in samples

5 In one example of the present invention, each hint sample describes one
transport packet. Each transport packet can be described as some amount of header
data, followed by some amount of payload from one media track. Since MPEG-2
transport packets are relatively small, a large number of hint samples may be
generated, and thus, these samples preferably should be as small as possible. Several
10 entries in the additional data table above may be used to minimize the size of samples,
but such factors may make some of the fields in the sample entries variable in size.

 If the 'otyp' entry in the data table has the value 0, indicating that payload data
is described in terms of media samples, hint samples may be of the following form
shown in Table 8:

Table 8 - Hint Sample Format Using Media Sample References

Length	Description
1	Track reference number of the media track holding the payload data for this packet. This can be mapped to a track ID using the 'tmap' entry in the additional data table. If the hint specifies 188 bytes of immediate data, this field is irrelevant.
1	The length of the immediate data for the packet. Note that this must be 188 or less, since transport packets are 188 bytes in length.
Variable	Bytes of immediate data to be used as the header for the transport packet. The number of bytes is described by the previous field.
Variable	The media sample number to use for the payload data. The default size of this field is 4 bytes, but may be modified by the presence of an 'msns' entry in the additional data table.
Variable	The media sample offset to use for the payload data. The default size of this field is 4 bytes, but may be modified by the presence of an 'msos' entry in the additional data table.

In one example of the present invention, it is not necessary to indicate the length of the payload data for the packet since in MPEG-2, this length is equal to 188 minus the size of the header data for the packet.

If the 'otyp' entry in the data table has the value 1, indicating that payload data is described in terms of file offsets, hint samples may be of the following form shown in Table 9:

Table 9

Length	Description
1	Track reference number of the media track holding the payload data for this packet. This can be mapped to a track ID using the 'tmap' entry in the additional data table. If the hint specifies 188 bytes of immediate data, this field is irrelevant.
1	The length of the immediate data for the packet. Note that this must be 188 or less since transport packets are 188 bytes in length.
Variable	Bytes of immediate data to be used as the header for the transport packet. The number of bytes is described by the previous field.
Variable	The file offset where the payload data is located. This offset is in the file where the data for the media track is located. The default size of this field is 4 bytes, but may be modified by the presence of an 'fosz' entry in the additional data table.

In one example of the present invention, hint samples may describe their offsets in terms of media samples or in terms of file offsets. Each of these has advantages and disadvantages. If hint samples specify payload in terms of media samples, they may be more resilient to additional editing of the file containing the media track, but may require additional processing for delivery. If hint samples specify payload in terms of file offsets, the payload data can be accessed relatively quickly, but any editing of the file containing the media track may invalidate the hints.

10

Appendix D — An example file

Provided below is a relatively short (six frame) sample file, with some of the relatively less important fields and objects left out (marked here by ellipsis "..."), and with some fictitious numbers to illustrate the overall structure of a file which is ready for streaming over RTP, according to one embodiment of the present invention. The media data has been left out; only the meta-data is shown.

15

```

moov -- the entire movie meta-data
mvhd -- overall movie information
...
TIME-SCALE          600
DURATION            2792
PREFERRED-RATE      1
VOLUME              255
MATRIX              [[1 0 0] [0 1 0] [0 0 1]]
...
NEXT-TRACK-ID       5 -- tracks 1 to 4 are here
trak -- this is the video track
tkhd
...
TRACK-ID            1
DURATION            2792
LAYER               0
...
MATRIX              [[1 0 0] [0 1 0] [0 0 1]]
WIDTH               176
HEIGHT              144
mdia
mdhd
...
TIME-SCALE          600
DURATION            2722
...
hdlr -- we use the basic video media handler
...
TYPE                mhlr
SUBTYPE             vide
MANUFACT            appl
...
NAME                Apple Video Media Handler
minf
vmhd
...
hdlr -- basic 'alias' disk data handler gets the data
...
TYPE                dhlr
SUBTYPE             alis
MANUFACT            appl
...
NAME                Apple Alias Data Handler
dinf
dref
...
ENTRY-COUNT         1
REFS                 [Pointer to this file]
stbl -- the complete sample table
stds -- the sample description(s)

```

-70-

```

...
ENTRY-COUNT          1
DESCRIPTIONS          [video sample description]
stts -- convert time to sample
...
ENTRY-COUNT          6
TIMETOSAMPLE          ((1 200) -- count, duration
                      (1 251)
                      (1 479)
                      (1 531)
                      (1 1022)
                      (1 239))
stss -- 'sync' or key sample numbers
...
ENTRY-COUNT          1
SYNCSAMPLES          (1)
stsc -- sample to chunk
...
ENTRY-COUNT          1
SAMPLETOCHUNK          ((1 1 1))
                      -- 1st chunk, samples/chunk, desc. number
stsz -- sample sizes
...
DEFSAMPLESIZE          0 -- no default size, all
different
ENTRY-COUNT          6
SAMPLESIZES          (664
                      616
                      1176
                      1304
                      2508
                      588)
stco -- chunk offsets into file
...
ENTRY-COUNT          6
CHUNKOFFSETS          (4743
                      5407
                      8010
                      12592
                      17302
                      25268)
trak -- this is the sound track
tkhd
...
TRACK-ID              2
DURATION              2792
...
VOLUME                1
...
mdia
mdhd

```

```

...
TIME-SCALE          8000
DURATION            37280
LANGUAGE             US English
...
hdlr -- handled by the basic sound handler
...
TYPE                mhlr
SUBTYPE              soun
MANUFACT             appl
...
NAME                Apple Sound Media Handler
minf
smhd
...
BALANCE              0
hdlr -- data fetched by usual disc data handler
...
TYPE                dhlr
SUBTYPE              alis
MANUFACT             appl
...
NAME                Apple Alias Data Handler
dinf
dref
...
ENTRY-COUNT          1
REFS                 [Pointer to this file]
stbl -- sample table for the sound
std -- sample descriptions
...
ENTRY-COUNT          1
DESCRIPTIONS          [Sound sample description, incl
GSM]
stts -- time to sample table
... -- sound is measured by uncompressed samples
ENTRY-COUNT          1
TIMETOSAMPLE         ((37280 1))
stsc
...
ENTRY-COUNT          2
SAMPLETOCHUNK         ((1 4000 1)
                      (10 1280 1))
-- first chunk, samples/chunk, desc. number
stsz
...
DEFSAMPLESIZE        1 -- all samples same size
ENTRY-COUNT          37280
stco -- chunk offset table
...
ENTRY-COUNT          10

```

-72-

```

                                CHUNKOFFSETS      (3093
                                                    3918
                                                    6023
                                                    9186
                                                    10915
                                                    13896 ...)
trak -- the RTP hints for the video track
tkhd
...
    TRACK-ID                    3
    DURATION                    2792
...
tref
    hint -- references the video track
        TRACKIDS                (1)
mdia
    mdhd
        ...
        TIME-SCALE              600
        DURATION                2792
        ...
    hdlr -- is 'played' by the hint media handler
        ...
        TYPE                    mhlr
        SUBTYPE                 hint
        MANUFACT                appl
        ...
        NAME                    hint media handler
minf
    gmhd
        ...
    hdlr -- if played, the regular disc handler would fetch data
        ...
        TYPE                    dhlr
        SUBTYPE                 alis
        MANUFACT                appl
        ...
        NAME                    Apple Alias Data Handler
dinf
    dref
        ...
        ENTRY-COUNT             1
        REFS                    [Pointer to this file]
stbl -- samples describe packets
    stsd
        ...
        ENTRY-COUNT             1
        DESCRIPTIONS            [hint sample description]
    stts -- one packet per frame for video
        ...
        ENTRY-COUNT             6
```


-73-

```
TIMETOSAMPLE      ((1 270)
                  (1 251)
                  (1 479)
                  (1 531)
                  (1 1022)
                  (1 239))
stss -- key sample derive from video
...
ENTRY-COUNT       1
SYNCSAMPLES       (1)
stsc -- sample to chunk table
...
ENTRY-COUNT       1
SAMPLETOCHUNK     ((1 1 1))
stsz -- sample sizes (packet instructions)
...
DEFSAMPLESIZE     0
ENTRY-COUNT       6
SAMPLESIZES       (52
                  52
                  52
                  52
                  102
                  52)
stco -- chunk offsets
...
ENTRY-COUNT       6
CHUNKOFFSETS      (6848
                  6900
                  10011
                  14721
                  20635
                  25856)
udta -- track is named for ease of identification
name
NAME              Hinted Video Track
trak -- the RTP hints for the sound track
tkhd
...
TRACK-ID          4
...
tref -- references the sound track
hint
TRACKIDS          (2)
mdia
mdhd
...
TIME-SCALE        8000
DURATION          37120
...
hdlr
```

```

...
TYPE                mhlr
SUBTYPE             hint
MANUFACT            appl
...
NAME                hint media handler
minf
  gmhd
  ...
  hdlr
    ...
    TYPE            dhlr
    SUBTYPE         alis
    MANUFACT        appl
    ...
    NAME            Apple Alias Data Handler
  dinf
    dref
      ...
      ENTRY-COUNT    1
      REFS            [Pointer to this file]
  stbl
    stsd
      ...
      ENTRY-COUNT    1
      DESCRIPTIONS    [hint sample description]
    stts -- time to sample
      ...
      ENTRY-COUNT    4
      TIMETOSAMPLE    ((1 960)
                       (7 4000)
                       (1 1120)
                       (1 7040))
    stsc
      ...
      ENTRY-COUNT    1
      SAMPLETOCHUNK    ((1 1 1))
    stsz
      ...
      DEFSAMPLESIZE    0
      ENTRY-COUNT    10
      SAMPLESIZES      (206
                       852
                       852
                       852
                       852
                       852 ...)
    stco
      ...
      ENTRY-COUNT    10
      CHUNKOFFSETS    (6952

```

-75-

7158
10063
11740
14773
16450 ...)

udta

NAME

Hinted Sound Track

udta
NAME
Hinted Sound Track